

FACULTATEA CALCULATOARE, INFORMATICA SI MICROELECTRONICA
UNIVERSITATEA TEHNICA A MOLDOVEI

MEDII INTERACTIVE DE DEZVOLTARE A PRODUSELOR SOFT
LUCRAREA DE LABORATOR#3

Web development

Autor:

Scripnic ALEXANDRU

lector asistent:

Irina COJANU

lector superior:

Svetlana COJOCARU

Laboratory work #2

1 Scopul lucrării de laborator

În această lucrare, se va seta un server pe calculatorul personal pentru lucrul cu un website personal.

Dezvoltarea unei aplicații web în HTML prin intermediul limbajului javascript și stilizarea lui prin intermediul la css.

2 Obiective

- Realizarea unui simplu Web Site personal
- Familiarizarea cu HTML si CSS
- Interactiuni Javascript

3 Laboratory work implementation

3.1 Tasks and Points

- Realizeaza un site cu folosirea maximala a tagurilor
- Pentru formatarea paginilor se va folosi CSS
- Site-ul trebuie sa pastreze toata informatia intr-o baza de date
- Site-ul trebuie sa contina AJAX Requests.
- Implimentarea XHR sau JSON responses. Careva din informatie trebuie sa fie dinamic incarcata pe pagina.

3.2 Analiza lucrarii de laborator

Repository link

Primul pas spre elaborarea task-urilor puse a fost setarea unui module bundler și anume webpack [1] cu scopul de a face un proces cât mai simplu și rapid de lucru cu dependențe externe și alte facilități care le voi descrie în continuare.

După setare configului de bază pentru webpack am instalat alte tooluri necesare pentru crearea unui website care sunt:

- babel-loader: care permite utilizarea celor mai noi versiuni a limbajului javascript - ES6[2]
- sass-loader: care permite scrieri CSS-ului într-o metodă cât mai optimală și rezervată sass[3] și includerea lui direct în index.html file la execuție pentru consumul minimal de resurse.

La fel webpack mi-a permis să fac un bundle proiectului meu, ce sporește viteza și consumul de resurse de către website.

Apoi am salvat ultima versiune stabilă de angular[4] framework pe bază căruia am și construit toată aplicația.

Al doilea pas în crearea aplicației a fost crearea structurii fișierelor pentru păstrarea directive-lor, controller-ilor, template-urilor și restul datelor.

Din start am determinat un routing pentru aplicație și am determinat de ce pagini voi avea nevoie. Apoi am început crearea aplicației statice într-un asemenea format, pentru a putea ușor adăuga conținut dinamic care va fi primit din partea serverului.

Pentru partea de UI am creat următoarele pagini care la rândul lor sunt completate dinamic:

- Home - unde este afișată lista de teme disponibile. reference:(3.1)
- Board (temă) - unde se afișează toate discuțiile pe o anumită temă reference:(3.2) și are forma de adăugare a unei discuții noi reference:(3.3)
- Thread (discuție) - unde se afișează toate comentariile la o discuție reference:(3.4) și adăugarea unui nou comentariu reference:(3.5)

Al treilea pas după ce am finisat partea statică a website-ului am început crearea serverului pe NodeJS[5] și ca framework de bază pentru ridicarea unui server pe dispozitiv am folosit ExpressJS[6] care mi-a dat posibilitatea de a crea rute virtuale pentru citirea imaginilor care vor fi pe partea serverului și un REST-API care mi-ar permite să creez posturi și threaduri în aplicație mea.

Cel mai greu pas a fost încărcarea unei imagini atât pe partea de client cât și pe partea serverului. Pentru a încărca o imagine pe server am folosit modulu exterior multer[7] care făcea handle la asemenea cazuri, iar pentru partea client am folosit o directive populară, destinată framework-ului angular, ng-file-upload[8] care la rândul său făcea handle la evenimente de genul upload.

După încărcarea fiecărei imagini am folosit modulul jimp[9], care permite editarea unei imagini, pentru a crea un preview, căci clientul să nu fie nevoit să fac download la toate imaginile în marimea lor originală, și le stocam aparte.

Al patrulea pas a fost adăugarea unei baze de date, care ar permite lucru ușor și rapid cu o cantitate mai mică de date. Am ales lokijs[10] care lucrează cu fișiere de tip JSON, ce imi este cel mai convinabil.

Și ultimul pas a fost conectarea părții client cu server, care am făcut-o prin intermediul directivei angular \$http[11] care la rândul său face apeluri externe de trimitere și primire a informației.

NOTĂ: Unica asemănare dintre 4chan și 3chan este design, în rest, javascript code, styles code și server-ul nu au nimic în comun, totul a fost creat de la zero. (nici html nu coincide).

3.3 Imagini



Figure 3.1 – Home Page

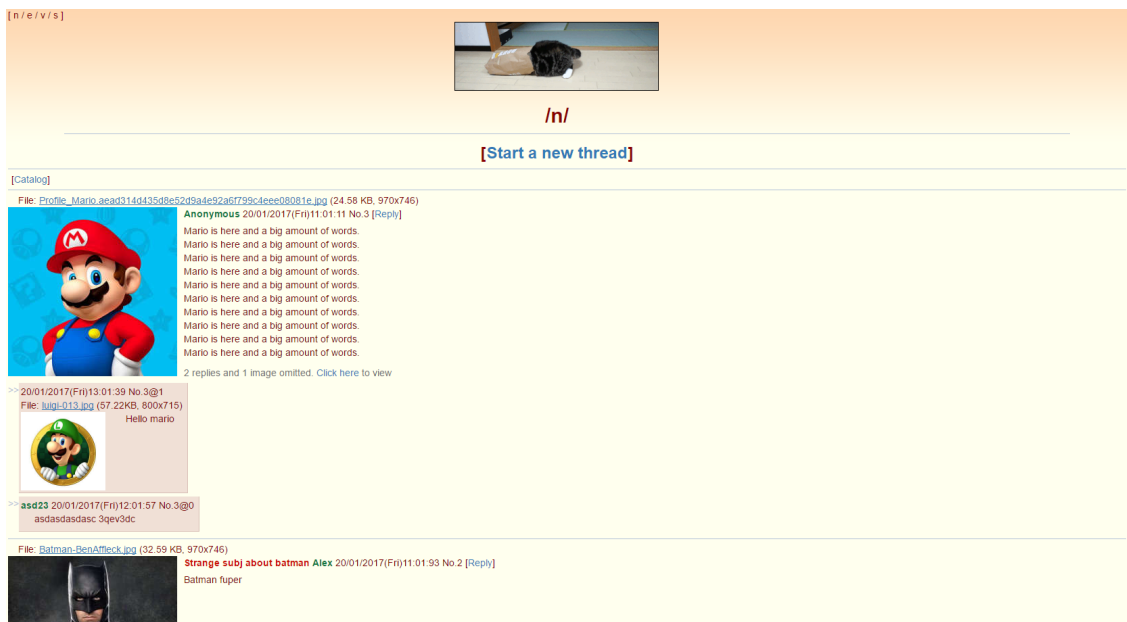


Figure 3.2– Board Page

Figure 3.3– New Thread Form

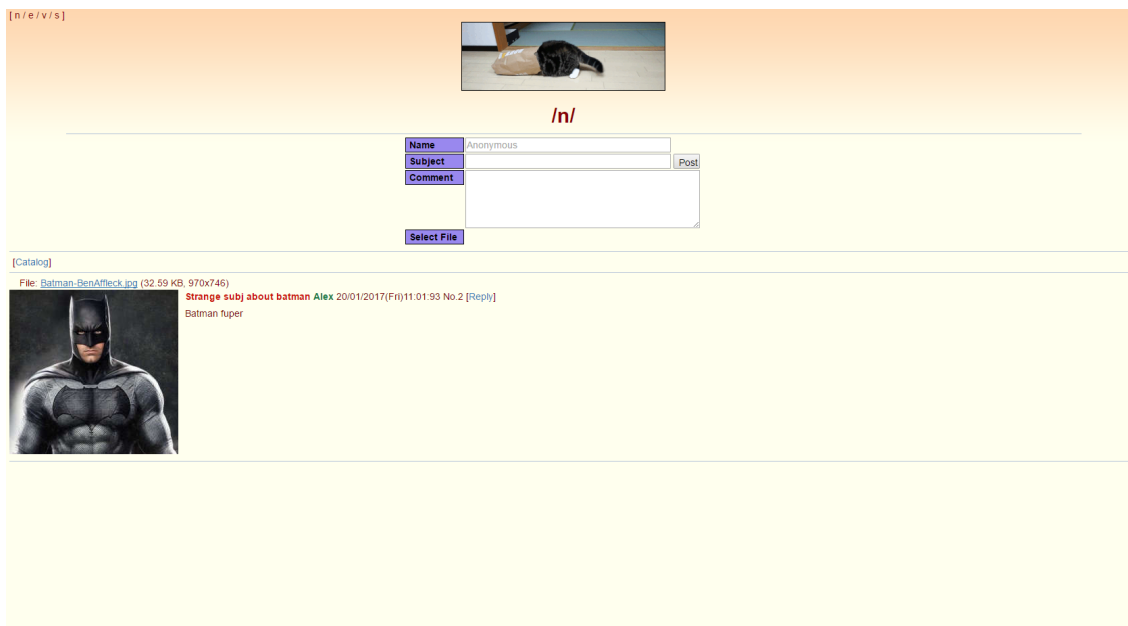


Figure 3.4– Thread Page



Figure 3.5– New post Form

Concluzie

În urma elaborării lucrării de laborator am avut însfârșit șansa de a învăța/perefecționa cunoștințele în domeniul stilizării unei aplicații web, mai ales folosind sass, care e mult mai comod și mai curat.

Am cunoscut module bundler-ul Webpack [1] care permite oferă o cantitate mare de posibilități și automatizări, ca exemplu folosirea ultimelor tehnologii ES6 și SASS, routing la dependențe externe automat în care este destul doar să instalezi prin intermediul package manager-ului npm un modul și el automat este inclus în proiect și cea mai deosebită funcție a Webpack-ului este un strat de security care nu permite variabile globale în proiect.

Cât pentru partea de backend am învățat ExpressJS[6] care oferă de asemenea o mulțime de posibilități ca crearea unui server, REST API, alte tipuri de API, conectarea la bază de date, routing și etc.

References

- 1 Webpack, *official page*, <https://webpack.github.io/>
- 2 EcmaScript6, <http://es6-features.org/>
- 3 Sass, *official page*, http://sass-lang.com/documentation/file.SASS_REFERENCE.html
- 4 Angular, *official page*, <https://angularjs.org/>
- 5 NodeJS, *official page*, <https://nodejs.org/en/>
- 6 ExpressJS, *official page*, <http://expressjs.com/>
- 7 Multer, *github page*, <https://github.com/expressjs/multer>
- 8 Lightweight Angular directive to upload files, *github page*, <https://github.com/danialfarid/ng-file-upload>
- 9 Jimp - An image processing library, *github page*, <https://github.com/oliver-moran/jimp>
- 10 LokiJs, *official page*, <http://lokijs.org/#/>
- 11 Angular \$http directive, *official page*, [\\$http](https://docs.angularjs.org/api/ng/service/$http)